Minimal Locked Trees

Brad Ballinger[1], David Charlton[2], Erik D. Demaine[3⋆], Martin L. Demaine[3], John Iacono[4], Ching-Hao Liu[5⋆⋆], and Sheung-Hung Poon[5⋆⋆]

[1] Davis School for Independent Study, 526 B Street, Davis, CA 95616, USA, ballingerbrad@yahoo.com
[2] Boston University Computer Science, 111 Cummington Street, Boston, MA 02135, USA, dcharlton@gmail.com
[3] MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar Street, Cambridge, MA 02139, USA, {edemaine,mdemaine}@mit.edu
[4] Department of Computer Science and Engineering, Polytechnic Institute of NYU, 5 MetroTech Center, Brooklyn NY 11201, USA, http://john.poly.edu
[5] Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan, chinghao.liu@gmail.com, spoon@cs.nthu.edu.tw

Abstract. Locked tree linkages have been known to exist in the plane since 1998, but it is still open whether they have a polynomial-time characterization. This paper examines the properties needed for planar trees to lock, with a focus on finding the smallest locked trees according to different measures of complexity, and suggests some new avenues of research for the problem of algorithmic characterization. First we present a locked linear tree with only eight edges. In contrast, the smallest previous locked tree has 15 edges. We further show minimality by proving that every locked linear tree has at least eight edges. We also show that a six-edge tree can interlock with a four-edge chain, which is the first locking result for individually unlocked trees. Next we present several new examples of locked trees with varying minimality results. Finally, we provide counterexamples to two conjectures of [12], [13] by showing the existence of two new types of locked tree: a locked orthogonal tree (all edges horizontal and vertical) and a locked equilateral tree (all edges unit length).

1 Introduction

A *locked tree* is a tree graph (linkage) embedded in the plane that is unable to reconfigure to some other configuration if we treat the edges as rigid bars that cannot intersect each other. The idea of locked trees goes back to 1997, in the context of an origami problem [8]. Only four main families of locked trees have been discovered so far. The first two locked trees, shown in Figure 1(a–b), were discovered soon after in 1998 [3]. In 2000, it was established that locked trees

⋆ Partially supported by NSF CAREER award CCF-0347776.
⋆⋆ Supported by grant 97-2221-E-007-054-MY3 of the National Science Council (NSC), Taiwan, R.O.C.

must have vertices of degree more than 2 (the Carpenter's Rule Theorem) [6, 14]. The third locked tree, shown in Figure 1c, shows that this result is tight: a single degree-3 vertex suffices to lock a tree [5]. The fourth locked tree, shown in Figure 1d, modified the first locked tree to reduce its graph diameter to 4, which is the smallest of any locked tree [12].

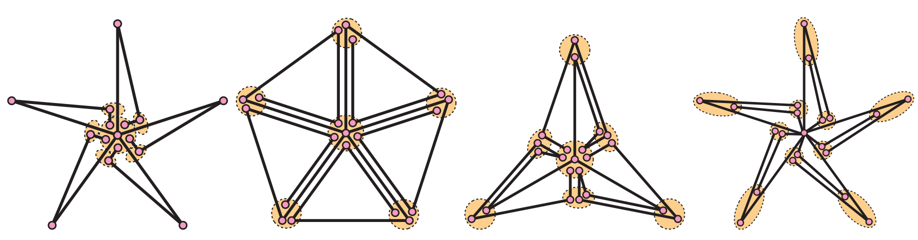
All four trees have a similar structure: they arrange repeated pieces in a cycle so that no piece can individually squeeze and so that no piece can individually expand without squeezing the other pieces (which in turn is impossible). Do all locked trees have this structure? This paper aims to find minimal examples of locked trees, with the goal of finding the "heart" of being locked. In particular we find smaller locked trees that lack the cyclic structure of previous examples.

It seems difficult to characterize locked trees. Toward this goal, some types of trees are easy to prove locked via recent algorithmic tools [5, 4], and we use this theory extensively here. On the other hand, deciding whether a tree linkage can be transformed from one configuration to another is PSPACE-complete [2]. However, this hardness result says nothing about the special case of testing whether a tree is locked. In the sections that follow, we describe several new examples and counterexamples in locked trees, and suggest ways in which they may hint at deeper results in the associated algorithmic theory.

Our results. We discover several new families of locked trees with several previously unobtained properties. We also introduce a new general category of locked tree, the *linear* locked tree, which in addition to being important for the study of locked linkages also provides an interesting special case for the algorithmic characterization of lockedness.

First, in Section 3, we present a locked tree with only eight edges. In contrast, the smallest previous locked tree is Figure 1a with 15 edges. Our tree is also the only locked tree other than Figure 1c that has just one degree-3 vertex (and the other degrees at most 2). Therefore we improve the number of edges in the smallest such tree from 21 to eight.

Our tree has the additional property that it is *linear*: its vertices lie (roughly) along a line (see the full definition in Section 3). In Section 4, we prove that all linear locked trees have at least eight edges, establishing minimality of our eight-edge locked tree. We conjecture further that all locked trees have at least eight edges, though this problem remains open.



(a) [3] **(b)** [3, tech. rep.] **(c)** [5] **(d)** [12]

Fig. 1: All previous families of locked trees. Shaded regions are tighter than drawn.

We also show in Section 4 that all linear locked trees have diameter at least 5. In Section 5, we find a linear locked tree of precisely this diameter, using nine edges, and further show that this is the smallest number of edges possible for a diameter-5 linear locked tree. In contrast, the (nonlinear) locked tree in Figure 1d has diameter 4, while no locked trees have diameter 3 [12].

Next we consider interlocked trees, in the spirit of interlocked 3D chains [9–11]. In Section 6, we show that though diameter 3 trees cannot lock, they can interlock. (In contrast, any number of diameter-2 trees cannot interlock, as they are star-shaped [7, 15, 10].) As a consequence, caterpillar trees, which generalize diameter-3 trees, can lock. Additionally, we prove for the first time that smaller trees suffice for interlocking: a six-edge tree can interlock with a four-edge chain.

Finally we solve two conjectures about the existence of locked trees with particular properties. On the easier side, we show in Section 7 that certain linear locked trees, such as our eight-edge locked tree, can be transformed to obtain locked orthogonal trees. Such trees were previously conjectured not to exist [13] because all examples in Figure 1 critically use angles strictly less than 90°.

Our technically most challenging result is the design of a locked equilateral tree, where every edge has the same length. The hexagonal analog of Figure 1b is tantalizingly close to this goal, as the edges can have lengths arbitrarily close to each other. But if the tree is to not overlap itself, the lengths cannot be made equal. For this reason, equilateral locked trees were conjectured not to exist [12]. Nonetheless, in Section 8, we find one. This result is quite challenging because previous algorithmic frameworks were unable to analyze the lockedness of trees with fixed edge lengths. Specifically, where previous locked trees were very tightly locked (within an arbitrarily small constant), our locked equilateral tree has fairly large positive gaps between edges, forcing us to carefully compute the freedom of motion instead of simply using topological limiting arguments.

2 Terminology

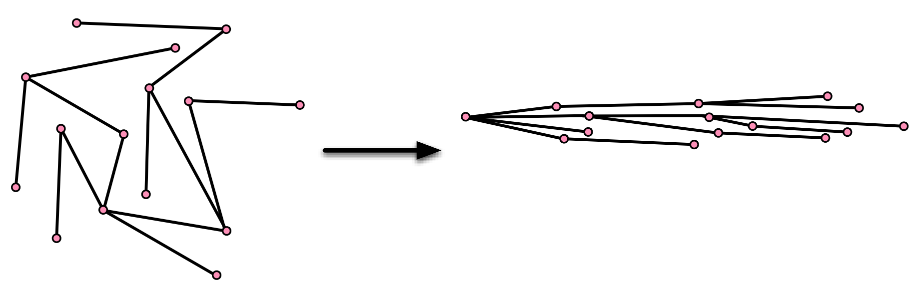


Fig. 2: Flattening a linkage: the initial tree (left) can be continuously transformed into the “flat” tree on the right, with all edges trailing off in the same direction from the root.

A *(planar) linkage* is a simple graph together with an assignment of a nonnegative real length to each edge and a combinatorial planar embedding (clockwise order of edges around each vertex and which edges form the outer face). A *con-*

figuration of a linkage is a (possibly self-intersecting) straight-line drawing of that graph in the plane, respecting the linkage's combinatorial embedding, such that the Euclidean distance between adjacent nodes equals the length assigned to their shared edge.

We are primarily interested in *nontouching* configurations, that is, configurations in which no edges intersect each other except at a shared vertex. The set of all such configurations is called the *configuration space* of the linkage. A *motion* of a nontouching configuration C is a continuous path in the configuration space beginning at C. A configuration of a tree linkage can be *flattened* if it has a motion transforming it as in Figure 2 so that all edges are trailing off in the same direction from an arbitrarily chosen root node. Otherwise, it is *unflattenable*. (Which node is chosen as the root does not affect the definition; see [3].) We say a tree configuration is *locked* if it is unflattenable, and a tree linkage is locked if it has a locked configuration.

To analyze nontouching configurations it is helpful to also consider *self-touching* configurations, where edges may overlap as long as they do not cross each other. This complicates the definitions, because edges can share the same geometric location. Care is also needed in generalizing the definition of a motion, because two geometrically identical configurations may have different sets of valid motions depending on the combinatorial ordering of the edges. A full discussion of these details is beyond our scope, so we rely on the formalization and results of [5], [4] and [1]. The reader who wishes for the intuition behind this theory can think of a self-touching configuration as a convergent sequence of nontouching configurations, but for the formal definitions see the references.

A self-touching configuration is *rigid* if it has no nonrigid motion. A configuration is *locked within* ε if no motion can change the position of any vertex by a distance of more than ε (modulo equivalence by rigid motions). A configuration C is *strongly locked* if, for any $\varepsilon > 0$, for any sufficiently small perturbation of C's vertices (respecting the original combinatorial relations between edges—see [5]), the resulting perturbed configuration is locked within ε. This property trivially implies unflattenability, and thus also that the underlying linkage is locked. Note that strongly locked configurations must be rigid and thus self-touching.

3 Minimal Locked Linear Tree

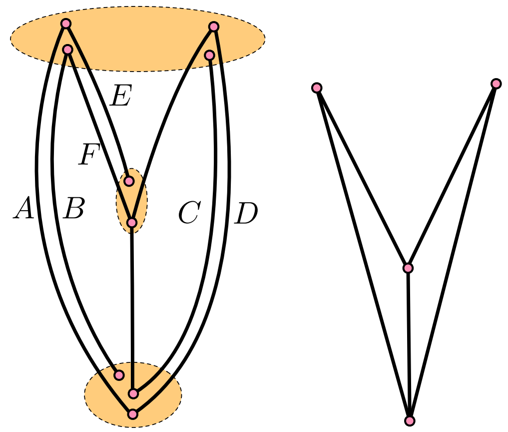
In this section we describe a new locked tree that is edge-minimal within an important class of locked trees, and is conjectured to be edge-minimal among all locked trees. Namely, a *linear configuration* is a (usually self-touching) configuration of a linkage in which all vertices lie on a single line. A *locked linear tree* is a tree linkage having an unflattenable linear configuration.

Note that our primary interest is still in nontouching configurations, but the existence of a linear configuration has implications for the general configuration space of the tree. Specifically, we make extensive use of the following lemma:

Lemma 1 (Theorem 8.1 from [5]). *Any rigid self-touching configuration is strongly locked.*

Because our proofs proceed by showing our linear trees rigid, this result implies that they remain locked even when the parameters are modified slightly to allow a nontouching configuration.

Consider the self-touching tree in Figure 3a. The linear geometry of this tree is a straight vertical line with only three distinct vertices at the top, center and bottom, but it is shown "pulled apart" to ease exposition. We claim this tree is rigid and thus, by Lemma 1, strongly locked. To show rigidity, we use two lemmas from [4]:



(a) Eight-bar locked linear tree. Shaded regions are tighter than drawn. All edges are straight lines, but shown "pulled apart".

(b) Reduced version of the tree after applying Lemma 2 and Lemma 3.

Fig. 3: The fewest-edge locked linear tree.

Lemma 2 (Rule 1 from [4]). *If a bar b is collocated with another bar b' of equal length, and two other bars incident to b' on each end form angles less than $90°$ on the same side as b, then any motion must keep b collocated with b' for some positive time.*

Lemma 3 (Rule 2 from [4]). *If a bar b is collocated with an incident bar b' of the same length whose other incident bar b'' forms a convex angle with b' surrounding b, then any motion must keep b collocated with b' for some positive time.*

Theorem 1. *The tree in Figure 3a is strongly locked.*

Proof: By Lemma 2, edges A and B must be collocated for positive time under any continuous motion, as must edges C and D. With these identifications, Lemma 3 shows that edges E and F must also remain collocated. We conclude that for positive time, the tree is equivalent to Figure 3b, which is trivially rigid. Therefore, the original tree is rigid and, by Lemma 1, strongly locked. □

4 Unfolding Linear Trees of Seven Edges

In Section 3, we presented a linear locked tree with eight edges. Now we will show that this is minimal: linear trees with at most seven edges can always be flattened. Because the full proof requires an extensive case analysis, we defer this to the full paper, and here present a sketch of how our arguments exploit the linearity of a tree.

Theorem 2. *A linear tree of diameter 4 can always be flattened.*

Lemma 4. *A linear tree of seven edges and diameter 5 can always be flattened.*

Lemma 5. *A linear tree of seven edges and diameter* 6 *can always be flattened.*

Proof sketch of Theorem 2, Lemma 4, Lemma 5: Because the tree's initial configuration lies on a line, many steps become simpler: first, if there are any loose edges along the perimeter of the tree, we can immediately straighten these. In Theorem 2, the tree has a center node, and we can then pivot all subtrees around that node so they lie in the same direction. This allows us to sequentially rotate out individual subtrees and straighten them one by one (a case analysis shows that if distinct subtrees are tangled together they can be safely pulled apart).

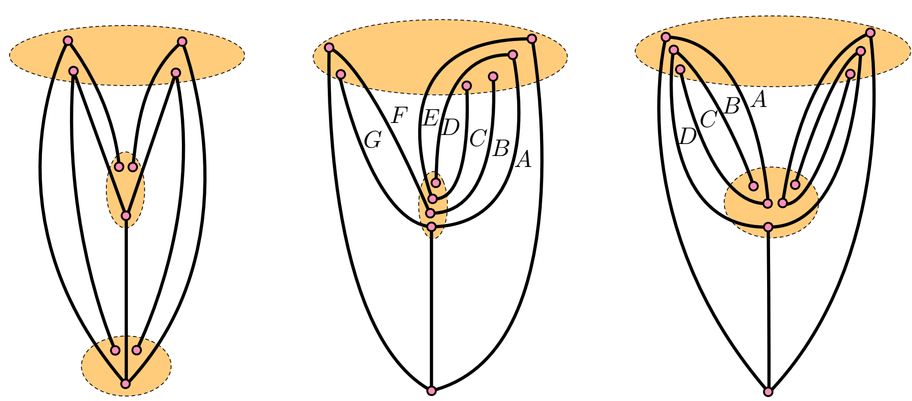
When the diameter is 5 or 6, the key observation is that the constraints do not allow a double-triangle structure as in Figure 3b. Specifically, case analysis shows the center edge cannot be formed, and thus the bounding quadrilateral can be expanded. When this quadrilateral becomes convex, the tree pulls apart easily. □

Because it was already shown in [6] that a seven-edge, diameter-7 tree (i.e., a 7-chain) cannot lock, combining these results immediately gives us the following:

Theorem 3. *A linear tree of at most seven edges can always be flattened.*

We thus conclude that the linear locked tree in Figure 3a has the fewest possible edges.

5 Additional Locked Linear Trees



(a) A locked tree having nine edges and the lowest diameter (5) of any possible locked linear tree.

(b) A 10-edge locked linear tree with a somewhat different structure. Edge labels appear to the right of their respective edge.

(c) Another symmetric locked linear tree, this time with 11 edges. Edge labels appear to the right of their respective edge.

Fig. 4: Additional locked linear trees.

Theorem 4. *The trees in Figure 4 are strongly locked.*

Like Theorem 1, all these theorems are proven by repeatedly applying Lemmas 2 and 3 until the configuration simplifies to Figure 3b, after which Lemma 1 applies.

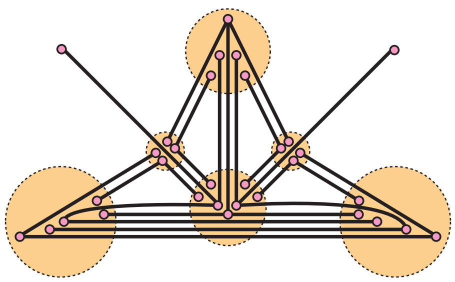
By a slight extension to the results of Section 4, we can prove the minimality of Figure 4a in a second sense, by showing that any diameter-5 linear locked tree requires at least nine edges:

Theorem 5. *A linear tree of* 8 *edges and of diameter* 5 *can always be flattened.*

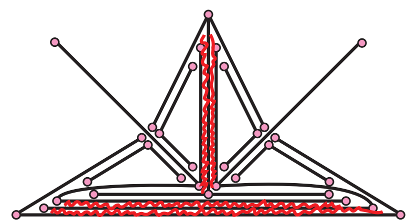
This claim is nearly implicit in the proof of Lemma 4; see the full paper.

6 Interlocked Trees

6.1 Diameter-3 Interlocked Trees. In this section we describe a set of eight interlocked trees of diameter 3 (although four of the "trees" are in fact 2-chains). Because diameter-2 trees cannot interlock (as they are star-shaped [7, 15, 10]), this example is tight. Because diameter-3 trees cannot lock, this is also the first example showing that the required diameter for interlocked (planar) trees is strictly below that of locked trees.



(a) Interlocked configuration (shaded regions are self-touching or very close).



(b) Identifications obtained from Lemma 2 and Lemma 3 (darkened areas indicate edges glued together).

Fig. 5: Eight interlocked diameter-3 trees.

For our proof, we introduce a new general lemma in the spirit of Lemma 2 and Lemma 3. The proof requires a geometric computation which we defer to the full version.

Lemma 6 ("Rule 3"). *If endpoints v_1 and v_3 of incident bars v_1v_2 and v_2v_3 are collocated with the endpoints of a third bar b, and bars incident to b form acute angles containing v_1 and v_3, then for positive time, any motion that moves v_1 or v_3 with respect to b must strictly increase the distance between v_2 and b.*

Theorem 6. *The eight diameter-*3 *trees in Figure 5a are strongly (inter)locked.*

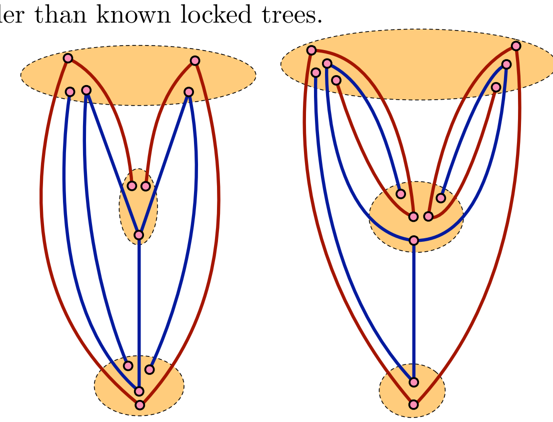
Proof: As with previous examples we begin by applying Lemma 2 and Lemma 3. The edge identifications from this process are shown in Figure 5b. It is enough to prove that the resulting figure is rigid, and the rest will follow from Lemma 1.

Now, observe that the 2-chains inside each of the four regions of the figure satisfy the requirements of Lemma 6, and that therefore the long diagonal edges are rigid: any rotation on their part would decrease the angular space allocated to some region, and push the center of the corresponding 2-chain closer to the opposing edge, contradicting the lemma. But then Lemma 6 implies the 2-chains themselves are glued in place for positive time.

The preceding leaves only the four loose edges around the outside. But because the 2-chains glue to their base vertices and to the long diagonals, Lemma 3 now applies, so these edges too are locked in place. □

6.2 Six-Edge Interlocked Tree. Here we describe a simple transformation that applies to many locked linear trees, yielding a smaller tree interlocked with a chain. Applying this transformation to Figure 4a, we obtain the smallest known instance of a tree interlocked with a chain. This is the first example of a planar interlocking tree strictly smaller than known locked trees.

Figure 6 shows the transformation. The basic idea is to disconnect a subtree at one end of the tree, replacing the connection with an extra edge that serves the same purpose, that is, such that the edge also constrains the subtree to remain adjacent to the same node. In Figure 6a, this gives us a 6-edge tree interlocked with a 4-edge chain, the best known.



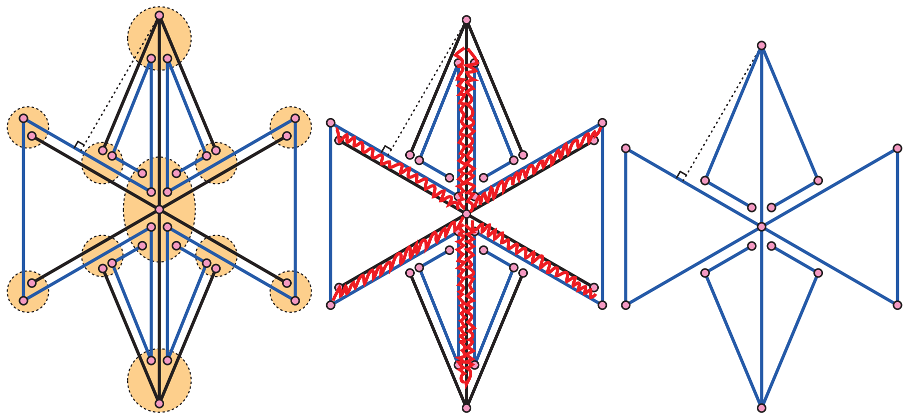
(a) Interlocked version of Figure 4a. **(b)** Interlocked version of Figure 4c.

Fig. 6: Interlocked variations of our locked trees.

Theorem 7. *The configurations in Figure 6a and Figure 6b are interlocked.*

As with their locked predecessors, successive applications of Lemma 2 and Lemma 3 suffice to prove rigidity of the configurations; see the full paper.

6.3 (Inter)locked Caterpillar. Here we describe an interesting example originally inspired by the search for the diameter-3 interlocked trees of Section 6.1. A *caterpillar graph* is a graph where removal of all leaf vertices and their incident edges results in a path. Because every vertex is at most one edge away from the central chain, the leaf vertices form the "legs" of the central chain "body" of the caterpillar. The intuition is of a graph that is locally low-diameter, or almost chain-like. Caterpillars provide a natural intermediate structure between collections of diameter-3 graphs and the full power of diameter 4, which was already known to lock.



(a) Interlocked caterpillar (black) and two 9-chains. **(b)** Identifications obtained from Lemma 2. **(c)** Simplified linkage after identifications and removal of extraneous bars.

Fig. 7: A locked caterpillar.

Because a caterpillar can take the place of any number of diameter-3 graphs, we can implicitly obtain a locked caterpillar directly from Figure 5a. However, in this section we describe a much simpler structure, and one that can be realized as the interlocking of a single ten-edge caterpillar and two 9-chains (or one 22-chain). We can also produce a single locking (not interlocking) caterpillar by merging the separate chains into the main body of the caterpillar.

Theorem 8. *The configuration in Figure 7a is rigid, and therefore strongly locked.*

This claim follows from successive applications of Lemma 2, Lemma 3 and Lemma 6, similar to Theorem 6.

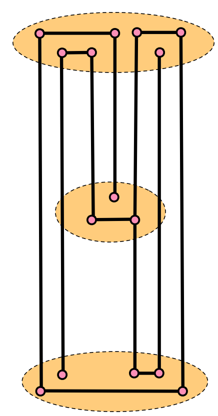


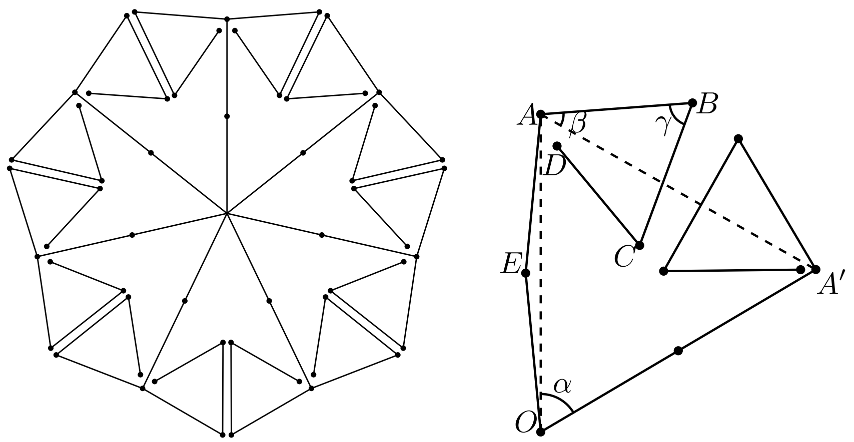
Fig. 8: Orthogonal version of Figure 3a.

7 Locked Orthogonal Tree

We now show that a simple transformation of the locked tree in Section 3 produces a locked orthogonal tree (a tree configuration such that all edges are axis-aligned), resolving a conjecture of Poon [13].

A modification to Figure 3a makes it orthogonal: see Figure 8. This diagram is still unflattenable (if the dimensions are chosen appropriately). The key is that this diagram can still be viewed as a small perturbation of the original tree if we add a zero-length edge to Figure 3a wherever Figure 8 has a horizontal edge, and thus we can again apply Lemma 1. Unfortunately, existing proofs of Lemma 1 do not work when the self-touching configuration has zero-length edges. It is a straightforward but technical matter to extend the lemma in this way. We defer the formal details to the full version.

8 Locked Equilateral Tree



(a) The locked unit tree, having seven arms of radius 2.

(b) Close-up of two adjacent arms. Roman letters refer to vertices, Greek to angles.

Fig. 9: A locked unit tree.

In [13], Poon conjectured that an equilateral tree (a tree linkage all of whose edges are equal length) could not lock. We provide a counterexample, shown in Figure 9a. This follows the "pinwheel" style of previous locked trees (Figure 1). The difference is that all previous locked trees (including the other examples in this paper) select their edge lengths so as to obtain an infinitesimally tight fit, whereas with unit edges we are limited to explicit numerical constraints.

Theorem 9. *The tree in Figure 9a is locked.*

Proof sketch: To prove lockedness in the absence of machinery derived from rigidity theory, we consider the angles and vertices labelled in Figure 9b. We claim that, under any continuous motion, the following invariants hold:

$$\sqrt{3} \le \|A - A'\| \le \sqrt{3} + 0.025 \quad (1) \qquad \left|\beta - \tfrac{\pi}{6}\right| \le 0.078\pi \quad (4)$$

$$1.94 \le \|A - O\| \quad (2) \qquad \tfrac{\pi}{3} \le \gamma \le \pi\left(\tfrac{1}{3} + 0.02\right) \quad (5)$$

$$0.2850989\pi \le \alpha \le 0.28941\pi \quad (3) \qquad \|C - AE\| \le 0.386 \quad (6)$$

We do so by showing that, if these inequalities hold for some valid configuration, then they actually hold strictly, that is, every instance of $\le$ above is actually $<$. Thus, these properties are preserved by any continuous motion. Due to space constraints, we give here the proofs for Equation 1 and Equation 3.

Consider Equation 3. The minimal α is attained when A and A' are at minimum distance from each other and maximum distance from the center vertex O. The latter is trivially 2, and the former is $\sqrt{3}$ by Equation 1. The angle so obtained is $2\arcsin(\frac{\sqrt{3}}{2\cdot 2}) > 0.2850989\pi$, as required. On the other hand, there are seven arms in the tree, and by the preceding that leaves $< 2\pi - 7(0.2850989)\pi$

of angular free space, and even if one pair of arms uses all of it, we still have $\alpha < 2\pi - 6(0.2850989)\pi < 0.28941\pi$, so Equation 3 holds strictly.

Now consider the distance $\|A - A'\|$ between two adjacent arms. If we look at the line from A to A', (4) and (5) show that it must pass through edge BC (and, by symmetry, that of the neighbor arm). The distance from A to BC is least when the three vertices form an equilateral triangle, in which case it is $\sqrt{3}/2$. Because this is true for both arms, and because the tree is not self-touching, the true distance between A and A' must be strictly greater than $\sqrt{3}$. On the other hand, by (3) the maximum angular distance between A and A' is 0.28941π. Given this, $\|A - A'\|$ is maximized when both vertices are at distance 2 from the center (it could be higher if one of the vertices approached the center very closely, but (2) prevents this; see below). In this case, the distance between them is $2 \cdot 2\sin(0.28941\pi/2) < \sqrt{3} + 0.25$, proving Equation 1 strict. □

9 Open Problems

The results of this paper open up several new questions, both in terms of their optimality and in exploring the newly discovered class of locked linear trees.

Figure 3a has eight edges. Is this the smallest possible for a locked (not necessarily linear) tree? We conjecture yes. We believe that a proof along the general outline of Theorem 3 may work, but the case analysis must be arranged more carefully in a general tree.

The orthogonal tree in Figure 8 has 14 edges. Is this minimal? We suspect so. A possible path to proving this conjecture is to show that any smaller orthogonal tree can be projected down to a locked linear tree with fewer than eight edges, contradicting Theorem 3.

Stefan Langerman proposed the idea of interlocked trees by asking whether multiple diameter-3 trees could interlock, which we have shown in Section 6 to be the case. However, this leaves minimality open: four diameter-3 trees can interlock with four 2-chains. Can this be done with fewer trees? Fewer edges?

Our results suggest some more general algorithmic questions. All of our locked trees were reduced to the two triangles of Figure 3b by repeated applications of Lemmas 2 and 3. This may not be a coincidence. Can every rigid linear tree can be reduced to a set of connected triangles by applying these lemmas, or simple extensions of them? In particular, we believe that rigidity in linear trees is a purely combinatorial (rather than geometric) property. Even more generally, is there an efficient algorithm to decide rigidity of linear trees? We suspect so.

In linear trees, there may also be a closer connection between rigidity and lockedness than is (known to be) true in general. If we start with a locked linear tree, and extend its loose edges until they are tightly constrained (and hence possibly satisfy the preconditions for Lemma 2 and Lemma 3), does the resulting graph have a rigid subtree? This is true for all the examples we are aware of, and may provide a starting point for an algorithmic characterization. Analysis of linear trees seems much more feasible than the general case. Is there an efficient algorithm to decide lockedness?

Acknowledgments. This work was initiated during a series of open-problem sessions for an MIT class on Geometric Folding Algorithms (6.885 in Fall 2007), and continued at the 23rd Bellairs Winter Workshop on Computational Geometry organized by Godfried Toussaint (February 2008). We thank the other participants of those sessions and workshops for providing a productive and inspiring environment. In particular, we thank Stefan Langerman for posing the problem of interlocked trees.

References

1. T. G. Abbott, E. D. Demaine, and B. Gassend. A generalized carpenter's rule theorem for self-touching linkages. Preprint, December 2007.
2. H. Alt, C. Knauer, G. Rote, and S. Whitesides. On the complexity of the linkage reconfiguration problem. In *Towards a Theory of Geometric Graphs*, volume 342 of *Contemporary Mathematics*, pages 1–14. AMS, 2004.
3. T. Biedl, E. D. Demaine, M. L. Demaine, S. Lazard, A. Lubiw, J. O'Rourke, S. Robbins, I. Streinu, G. Toussaint, and S. Whitesides. A note on reconfiguring tree linkages: Trees can lock. *Discrete Applied Mathematics*, 117(1–3):293–297, 2002. The full paper is Technical Report SOCS-00.7, School of Computer Science, McGill University, September 2000. Originally appeared at CCCG 1998.
4. R. Connelly, E. D. Demaine, M. L. Demaine, S. Fekete, S. Langerman, J. S. B. Mitchell, A. Ribó, and G. Rote. Locked and unlocked chains of planar shapes. In *Proceedings of the 22nd Annual ACM Symposium on Computational Geometry*, pages 61–70, Sedona, Arizona, June 2006.
5. R. Connelly, E. D. Demaine, and G. Rote. Infinitesimally locked self-touching linkages with applications to locked trees. In *Physical Knots: Knotting, Linking, and Folding of Geometric Objects in* $\mathbb{R}^3$, pages 287–311. AMS, 2002.
6. R. Connelly, E. D. Demaine, and G. Rote. Straightening polygonal arcs and convexifying polygonal cycles. *Discrete & Computational Geometry*, 30(2):205–239, 2003.
7. R. Dawson. On removing a ball without disturbing the others. *Mathematics Magazine*, 57(1):27–30, 1984.
8. E. D. Demaine and M. L. Demaine. Computing extreme origami bases. Technical Report CS-97-22, Dept. of Computer Science, University of Waterloo, May 1997.
9. E. D. Demaine, S. Langerman, J. O'Rourke, and J. Snoeyink. Interlocked open linkages with few joints. In *Proceedings of the 18th Annual ACM Symposium on Computational Geometry*, pages 189–198, Barcelona, Spain, June 2002.
10. E. D. Demaine, S. Langerman, J. O'Rourke, and J. Snoeyink. Interlocked open and closed linkages with few joints. *Computational Geometry: Theory and Applications*, 26(1):37–45, August 2003.
11. J. Glass, B. Lu, J. O'Rourke, and J. K. Zhong. A 2-chain can interlock with an open 11-chain. *Geombinatorics*, 15(4):166–176, April 2006.
12. S.-H. Poon. On straightening low-diameter unit trees. In *Revised Papers from the 13th International Symposium on Graph Drawing*, LNCS 3843, pages 519–521, Limerick, Ireland, September 2005.
13. S.-H. Poon. On unfolding lattice polygons/trees and diameter-4 trees. In *Proceedings of the 12th Annual International Computing and Combinatorics Conference*, pages 186–195, 2006.
14. I. Streinu. Pseudo-triangulations, rigidity and motion planning. *Discrete & Computational Geometry*, 34(4):587–635, November 2005.
15. G. T. Toussaint. Movable separability of sets. In *Computational Geometry*, pages 335–375. North-Holland, Amsterdam, 1985.